The boa Package

April 18, 2007

Version 1.1.6-1

Date 2007-04-18

Title Bayesian Output Analysis Program (BOA) for MCMC

Author Brian J. Smith

Maintainer Brian J. Smith <brian-j-smith@uiowa.edu>

Depends R (>= 1.7)

Description A menu-driven program and library of functions for carrying out convergence diagnostics and statistical and graphical analysis of Markov chain Monte Carlo sampling output.

License GPL version 2 or newer

URL http://www.public-health.uiowa.edu/boa

R topics documented:

boa.acf . . . 3
boa.batchMeans . . . 3
boa.chain . . . 4
boa.chain.add . . . 5
boa.chain.collapse . . . 6
boa.chain.del . . . 6
boa.chain.eval . . . 7
boa.chain.gandr . . . 7
boa.chain.import . . . 9
boa.chain.info . . . 10
boa.chain.reset . . . 11
boa.chain.subset . . . 11
boa.chain.support . . . 12
boa.getiter . . . 13
boa.getparms . . . 13
boa.geweke . . . 14
boa.gewekePwr . . . 15

boa.handw . . . 15
boa.hpd . . . 16
boa.importASCII . . . 17
boa.importBUGS . . . 18
boa.importMatrix . . . 18
boa.init . . . 19
boa.iter . . . 20
boa.license . . . 20
boa.load . . . 21
boa.menu . . . 21
boa.par . . . 22
boa.pardesc . . . 24
boa.plot . . . 25
boa.plot.acf . . . 26
boa.plot.bandg . . . 27
boa.plot.close . . . 28
boa.plot.density . . . 28
boa.plot.gandr . . . 29
boa.plot.geweke . . . 30
boa.plot.history . . . 31
boa.plot.open . . . 32
boa.plot.par . . . 32
boa.plot.title . . . 33
boa.plot.trace . . . 33
boa.pnames . . . 34
boa.print.acf . . . 34
boa.print.cor . . . 35
boa.print.gandr . . . 35
boa.print.geweke . . . 36
boa.print.handw . . . 36
boa.print.hpd . . . 37
boa.print.info . . . 38
boa.print.par . . . 38
boa.print.randl . . . 39
boa.print.stats . . . 40
boa.quit . . . 40
boa.randl . . . 41
boa.save . . . 42
boa.sortiter . . . 42
boa.sortparms . . . 43
boa.stats . . . 44
boa.transform . . . 44
boa.version . . . 45
colVars . . . 46
line . . . 46

Index **47**

boa.acf *Autocorrelation Function*

Description

Computes lag autocorrelations for the parameters in an MCMC sequence.

Usage

```
boa.acf(link, lags)
```

Arguments

link	Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to `dimnames(link)`
lags	Vector of lags at which to estimate the autocorrelation function.

Value

A matrix whose columns and rows contain the estimated autocorrelation functions at the specified lags and the monitored parameters, respectively.

Author(s)

Brian J. Smith

See Also

boa.plot, boa.plot.acf, boa.print.acf

boa.batchMeans *Batch Means*

Description

Computes batch means for the parameters in an MCMC sequence.

Usage

```
boa.batchMeans(link, size)
```

Arguments

link	Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to `dimnames(link)`.
size	Number of iterations to include in each batch.

Value

A matrix whose columns and rows contain the means within sequential batches and the monitored parameters, respectively.

Author(s)

Brian J. Smith

`boa.chain` *Session Lists of MCMC Sequences*

Description

Extracts and sets the MCMC sequence(s) stored internally for use during a BOA session. Users can safely use this function to extract data. However, it should not be called directly to modify the stored MCMC sequence(s). Instead, data management should accomplished with the libboa.chain functions.

Usage

```
boa.chain(...)
```

Arguments

`...`	A list may be given as the only argument, or a character string given as the only argument, or any number of arguments may be in the `<name> = <value>` form, or no argument at all may be given. See the Value and Side Effects sections for further details.

Value

If no arguments are supplied, a list of the current values is returned. If a character string is given, the current value of the named variable is returned. Otherwise, no values are returned.

Parameter Names and Default Values

master = list() Master list containing all added MCMC sequences. Sequences may only be added to or deleted from the master list. The sequences in this list are never modified or used in any of the analyses. It serves as a template for the working list described below. The parameters and iterations may differ between the sequences. Each element in the list is a matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names are stored in the dimnames.

master.support = list() List containing the support for each of the parameters in 'master'. Each element in the list is a matrix whose columns and rows contain the monitored parameters and the support (lower and upper limits), respectively. The default for each parameter is c(-Inf, Inf).

work = master Working list used in all analyses. This list is essentially a modifiable copy of 'master'. All data management functions operate on the working list.

work.support = master.support List containing the support for each of the parameters in 'work'.

work.sync = TRUE Logical value indicating that the working list is an identical copy of the master list. This variable is automatically set to F after a successful call to `boa.chain.subset` or `boa.chain.collapse`. While `work.sync = F`, only the master list will be updated if MCMC sequences are added or deleted. A call to `boa.chain.reset` will reset this variable to T and copy the master list to the working list.

Side Effects

When variables are set, boa.chain() modifies the internal list `.boa.chain`. If boa.chain() is called with either a list as the single argument, or with one or more arguments in the `<name> = <value>` form, the variables specified by the names in the arguments are modified.

Author(s)

Brian J. Smith

`boa.chain.add` *Add MCMC Sequence*

Description

Adds an MCMC sequence to the session lists of sequences. Most users should not call this function directly. Passing improperly formatted data to this function will most likely break the functions in this library. The preferred method of adding a new sequence is to call the function `boa.chain.import`.

Usage

```
boa.chain.add(link, lname)
```

Arguments

`link`	Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to `dimnames(link)`.
`lname`	Character string giving the list name to use for the newly added MCMC sequence.

Value

A logical value indicating that 'link' has been successfully added.

Author(s)

Brian J. Smith

See Also

boa.chain.import

---

boa.chain.collapse *Concatenate MCMC Sequences*

Description

Concatenates together all of the MCMC sequences in the working session list of sequences. Only those parameters common to all sequences are kept.

Usage

```
boa.chain.collapse()
```

Value

A logical value indicating that the MCMC sequences have been collapsed. If the sequences share no common parameters, no modifications are made to the working list and FALSE is returned.

Author(s)

Brian J. Smith

See Also

codeboa.chain.reset, codeboa.chain.subset

---

boa.chain.del *Delete MCMC Sequences*

Description

Delete MCMC sequences from the session list of sequences.

Usage

```
boa.chain.del(lnames, pnames)
```

Arguments

lnames	Character vector giving the names of the MCMC sequences in the session list of sequences to be deleted. If omitted, no sequences are deleted.
pnames	Character vector giving the names of the parameters in the MCMC sequences to be deleted. If omitted, no parameters are deleted.

Side Effects

The specified MCMC sequences are deleted from the session lists of sequences.

Author(s)

Brian J. Smith

---

boa.chain.eval *Formulate New Parameter*

Description

Evaluates a user-specified expression to formulate a new parameter in each of the system lists of MCMC sequences.

Usage

```
boa.chain.eval(expr, pname)
```

Arguments

expr	S expression object used to formulate the new parameter. Typically, the new parameter is a function of the existing parameters.
pname	Name given to the new parameter.

Side Effects

The specified parameter is added to the session lists of MCMC sequences.

Author(s)

Brian J. Smith

---

boa.chain.gandr *Gelman and Rubin Convergence Diagnostics*

Description

Computes the Gelman and Rubin convergence diagnostics for a list of MCMC sequences. Estimates are calculated from the second half of each sequence.

Usage

```
boa.chain.gandr(chain, chain.support, alpha, pnames, window, to)
```

Arguments

chain	List of matrices whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to the dimnames.
chain.support	List of matrices whose columns and rows contain the monitored parameters and the support (lower and upper limits), respectively.
alpha	Quantile (1 - alpha / 2) at which to estimate the upper limit of the shrink factor.
pnames	Character vector giving the names of the parameters to use in the analysis. If omitted, all parameters are used.
window	Proportion of interations to include in the analysis. If omitted, 50% are included.
to	Largest iteration to include in the analysis. If omitted, no upper bound is set.

Value

psrf	A vector containing the Gelman and Rubin (uncorrected) potential scale reduction factors for the monitored parameters.
csrf	A matrix whose columns and rows are the Gelman and Rubin corrected scale reduction factors (i.e. shrink factor estimates at the median and specified quantile of the sampling distribution) and the monitored parameters, respectively. A correction of (df + 3) / (df + 1) is applied to the scale reduction factors.
mpsrf	A numeric value giving the multivariate potential scale reduction factor proposed by Brooks and Gelman.
window	A numeric vector with two elements giving the range of the iterations used in the analysis.

Author(s)

Brian J. Smith, Nicky Best, Kate Cowles

References

1. Brooks, S. and Gelman, A. (1998). General methods for monitoring convergence of iterative simulations. Journal of Computational and Graphical Statistics, 7(4), 434-55.

2. Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. Statistical Science, 7, 457-72.

See Also

boa.plot, boa.plot.bandg, boa.plot.gandr, boa.print.gandr

`boa.chain.import` *Import MCMC Sequence*

Description

Reads an MCMC sequence from an external file and adds it to the session lists of sequences.

Usage

```
boa.chain.import(prefix, path = boa.par("path"), type = "ASCII")
```

Arguments

`prefix`	Character string giving the prefix for the file(s) in which the MCMC sequence is stored.
`path`	Character string giving the directory path in which the file(s) subsides.
`type`	Character string specifying the type of data to be imported. Currently, the supported types are: • "ASCII" : ASCII text file. Calls `boa.importASCII` to import data from "prefix.txt". • "BUGS" : BUGS output files. Calls `boa.importBUGS` to import data from "prefix.ind" and "prefix.out". • "S" : S data frame or matrix.

Value

A logical value indicating that the specified file(s) has been successfully imported.

Side Effects

The imported MCMC sequence is automatically added to the session lists of sequences via a call to `boa.chain.add`.

Author(s)

Brian J. Smith

See Also

`boa.chain.add`, `boa.importASCII`, `boa.importBUGS`

`boa.chain.info` *Information about MCMC Sequences*

Description

Returns summary information from a list of MCMC sequences.

Usage

```
boa.chain.info(chain, chain.support)
```

Arguments

`chain`	List of matrices whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to the dimnames.
`chain.support`	List of matrices whose columns and rows contain the monitored parameters and the support (lower and upper limits), respectively.

Value

`lnames`	Character vector giving the names of the MCMC sequences in the session list of sequences.
`pnames`	List of character vectors giving the parameter names in each of the MCMC sequences.
`iter`	List of numeric vectors giving the iterations from each MCMC sequence.
`iter.range`	Matrix whose columns give the range of the iterations for the MCMC sequences named in the rows.
`support`	List of numeric vectors giving the support for the parameters in each of the MCMC sequences.

Author(s)

Brian J. Smith

See Also

`boa.print.info`

boa.chain.reset *Reset MCMC Sequences*

Description

Returns the data in the working list of MCMC sequences to the state it was in when originally imported. This function undoes any subsetting or collapsing that was done via the boa.chain.collapse or boa.chain.subset functions.

Usage

```
boa.chain.reset()
```

Side Effects

Returns the data in the working list of MCMC sequences to the state it was in when originally imported. This function undoes any subsetting or collapsing that was done via the boa.chain.collapse or boa.chain.subset functions.

Author(s)

Brian J. Smith

See Also

boa.chain.collapse, boa.chain.subset

boa.chain.subset *Subset MCMC Sequences*

Description

Selects a subset of the MCMC sequences stored in the working session list of sequences.

Usage

```
boa.chain.subset(lnames, pnames, iter)
```

Arguments

lnames	Character vector giving the names of the MCMC sequences to include in the subset. If omitted, all sequences are included.
pnames	Character vector giving the names of the parameters to include in the subset. If omitted, all parameters are included.
iter	Character vector giving the names of the parameters to include in the subset. If omitted, all parameters are included.

Value

A logical value indicating that the data has been successfully subsetted. If the requested subset contains no data, the working session list is not modified and FALSE is returned.

Side Effects

The subsetted data, if not an empty set, is copied to the working session list of MCMC sequences used in all analyses.

Author(s)

Brian J. Smith

See Also

boa.chain.collapse, boa.chain.reset

---

boa.chain.support *Set Support for Parameters*

Description

Modifies the support (range of possible values) for the parameters in the session lists of MCMC sequences.

Usage

```
boa.chain.support(lnames, pnames, limits)
```

Arguments

lnames	Character vector giving the names of the MCMC sequences over which to apply the changes. If omitted, changes are applied to all sequences.
pnames	Character vector giving the names of the parameters whose support is to be modified.
limits	Numeric vector with two elements giving the lower and upper limits, respectively, of the support for the specified parameters. Unbounded lower or upper limits should be specified as -Inf or Inf, respectively.

Value

A logical vector with two elements indicating that the support has been changed for variables in the master list and working list of MCMC sequences, respectively.

Side Effects

The support for each of the specified parameters is changed in the session lists of MCMC sequences.

Author(s)

Brian J. Smith

`boa.getiter` *Extract Iterations*

Description

Extracts a subset of the iterations in an MCMC sequence.

Usage

```
boa.getiter(link, iter)
```

Arguments

`link`	Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to `dimnames(link)`.
`iter`	Numeric vector containing the iterations to extract from 'link'.

Value

A subset of 'link' containing all prior parameters and the iterations specified in 'iter'.

Author(s)

Brian J. Smith

See Also

`boa.iter`, `boa.getparms`

`boa.getparms` *Extract Parameters*

Description

Extracts a subset of the parameters in an MCMC sequence.

Usage

```
boa.getparms(link, pnames)
```

Arguments

link	Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to dimnames(link).
pnames	Character vector containing the names of the parameters to extract from 'link'.

Value

A subset of 'link' containing the specified parameters in 'pnames' and all prior iterations.

Author(s)

Brian J. Smith

See Also

boa.getiter, boa.pnames

---

boa.geweke *Geweke Convergence Diagnostics*

Description

Computes the Geweke convergence diagnostics for the parameters in an MCMC sequence.

Usage

```
boa.geweke(link, p.first, p.last)
```

Arguments

link	Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to dimnames(link).
p.first	Proportion of iterations to include in the first window.
p.last	Proportion of iterations to include in the last window.

Value

A matrix whose columns contain the Z-Scores and associated p-values and whose rows contain the monitored parameters.

Author(s)

Brian J. Smith

References

Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to calculating posterior moments. In Bayesian Statistics 4, (ed. J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith). Clarendon Press, Oxford, UK.

See Also

`boa.plot`, `boa.plot.geweke`, `boa.print.geweke`

`boa.gewekePwr` *Geweke Spectral Density Estimation*

Description

Estimates the spectrum frequency for the parameters in an MCMC sequence.

Usage

```
boa.gewekePwr(link)
```

Arguments

`link`	Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to `dimnames(link)`.

Value

A vector of spectrum frequencies for the monitored parameters.

Author(s)

Brian J. Smith

`boa.handw` *Heidelberger and Welch Convergence Diagnostics*

Description

Computes the Heidleberger and Welch convergence diagnostics for the parameters in an MCMC sequence.

Usage

```
boa.handw(link, error, alpha)
```

Arguments

`link`	Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to `dimnames(link)`.
`error`	Accuracy of the posterior estimates for the parameters.
`alpha`	Alpha level for the confidence in the sample mean of the retained iterations.

Value

A matrix whose columns and rows are the Heidleberger and Welch convergence diagnostics (i.e. stationarity test, number of iterations to keep and to drop, Cramer-von-Mises statistic, halfwidth test, mean, and halfwidth) and the monitored parameters, respectively.

Author(s)

Brian J. Smith, Nicky Best, Kate Cowles

References

Heidelberger, P. and Welch, P. (1983). Simulation run length control in the presence of an initial transient. Operations Research, 31, 1109-44.

See Also

`boa.print.handw`

`boa.hpd` *Highest Probability Density Interval*

Description

Estimates the highest probability density (HPD) interval for the given parameter draws. Uses the Chen and Shao algorithm assuming a unimodal marginal posterior distribution.

Usage

```
boa.hpd(x, alpha)
```

Arguments

`x`	MCMC draws from the marginal posterior to use in computing the HPD.
`alpha`	Specifies the 100*(1 - alpha)% interal to compute.

Value

A vector containing the lower and upper bound of the HPD interval, labeled "Lower Bound" and "Upper Bound", respectively.

Author(s)

Brian J. Smith

References

Chen, M-H. and Shao, Q-M. (1999). Monte Carlo estimation of Bayesian credible and HPD intervals. Journal of Computational and Graphical Statistics, 8(1), 69-92.

---

boa.importASCII *Import ASCII Data*

Description

Import data from an ASCII text file. The variables and iterations should appear in the columns and rows, respectively, of the file. The variable names must be given in the first row. The iteration numbers are taken from the the column entitled "iter" (case-sensitive), if it exists; otherwise, the numbers 1..n are used, where n is the number of rows. The columns may be separated by white space or tabs.

Usage

```
boa.importASCII(prefix, path = NULL)
```

Arguments

`prefix`	Character string giving the prefix for the files in which the ASCII data is stored. boa.importASCII() looks for the file "prefix.txt".
`path`	Character string giving the directory path in which the file subsides. This argument may be omitted if the file is located in the current working directory. The specified path should not end with a slash(es).

Value

If the data is successfully imported, a matrix is returned whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names are stored in the dimnames of the returned matrix. Otherwise, NULL is returned.

Author(s)

Brian J. Smith

See Also

`boa.chain.import`, `boa.importBUGS`, `boa.importMatrix`

boa.importBUGS *Import BUGS Data*

Description

Import data from BUGS output files.

Usage

```
boa.importBUGS(prefix, path = NULL)
```

Arguments

prefix	Character string giving the prefix for the files in which the BUGS output is stored. boa.importBUGS() looks for the two files "prefix.ind" and "prefix.out".
path	Character string giving the directory path in which the file subsides. This argument may be omitted if the file is located in the current working directory. The specified path should not end with a slash(es).

Value

If the data is successfully imported, a matrix is returned whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names are stored in the dimnames of the returned matrix. Otherwise, NULL is returned.

Author(s)

Brian J. Smith

See Also

boa.chain.import, boa.importASCII, boa.importMatrix

boa.importMatrix *Import ASCII Data*

Description

Import data from an S numeric matrix object. The variables and iterations should appear in the columns and rows, respectively, of the matrix. The variable names and iteration numbers may be optionally specified in the dimnames of the matrix object. Otherwise, the iterations will be numbered 1..n, where n is the number of rows.

Usage

```
boa.importMatrix(prefix)
```

Arguments

prefix	Character string giving the name of the S object.

Value

If the data is successfully imported, a matrix is returned whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names are stored in the dimnames of the returned matrix. Otherwise, NULL is returned.

Author(s)

Brian J. Smith

See Also

boa.chain.import, boa.importASCII, boa.importBUGS

---

boa.init *Start Session*

Description

boa.init() is the first function called to start a BOA session when using the command-line. It sets up the internal data structures and initializes them with the appropriate default values.

Usage

```
boa.init(recover = FALSE)
```

Arguments

recover	a logical value for use if the previous BOA menu session terminated unexpectedly. It may be possible to crash the menu system by supplying the wrong type of data. In the event of a crash, `recover = TRUE` may be used to ensure that no data is lost.

Author(s)

Brian J. Smith

See Also

boa.load, boa.quit

boa.iter *Extract Iteration Numbers*

Description

Extracts the iteration numbers from an MCMC sequence.

Usage

```
boa.iter(link)
```

Arguments

link	Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to `dimnames(link)`.

Value

A numeric vector containing the iteration numbers in 'link'.

Author(s)

Brian J. Smith

See Also

boa.getiter, boa.pnames

boa.license *Licensing Information*

Description

Prints the licensing information for the Bayesian Output Analysis Program (BOA).

Usage

```
boa.license()
```

Author(s)

Brian J. Smith

See Also

boa.version

boa.load *Load Session*

Description

Loads the data and global parameter settings from a previously saved BOA session.

Usage

```
boa.load(name, envir = globalenv())
```

Arguments

name	Character string giving the name of the object containing the session to be loaded.
envir	The 'environment' from which the object should come. For more information, consult the help documentation in R on the `get` function.

Value

A logical value indicating that the object was successfully loaded.

Author(s)

Brian J. Smith

boa.menu *Interactive BOA Menu*

Description

Starts the interactive menu for BOA. Provides a menu-driven interface to all of the convergence diagnostics and statistical and graphical analysis tools.

Usage

```
boa.menu(recover = FALSE)
```

Arguments

recover	a logical value for use if the previous BOA menu session terminated unexpectedly. It may be possible to crash the menu system by supplying the wrong type of data. In the event of a crash, `recover = TRUE` may be used to ensure that no data is lost.

Warning

Do not use the `recover` argument unless starting the menu immediately after a crash.

Note

`boa.menu` is the only function in the BOA library that need be called in order to use the menu-driven interface.

Author(s)

Brian J. Smith

---

boa.par *Global Parameters*

Description

Displays and sets the global parameters stored internally for use during a BOA session.

Usage

```
boa.par(...)
```

Arguments

`...`	A list may be given as the only argument, or a character string given as the only argument, or any number of arguments may be in the `<name> = <value>` form, or no argument at all may be given.

Value

If no arguments are supplied, a list of the current values is returned. If a character string is given, the current value of the named variable is returned. Otherwise, a list of the named variables and their old values is returned, but not printed.

Parameter Names and Default Values

acf.lags = c(1, 5, 10, 50) Numeric vector of lags at which to estimate the autocorrelation function.

alpha = 0.05 Type I error rate used for all statistical tests and confidence intervals.

ASCIIext = ".txt" Character string giving the filename extension used when importing ASCII files.

bandwidth = function(x) 0.5 * diff(range(x)) / (log(length(x)) + 1) Function for computing the bandwidth used in estimating the density functions for parameters. This should take one argument which is a numeric vector of data on which density estimation is to be performed. A constant bandwidth may be specified by having this function return the desired constant.

batch.size = 50 Number of iterations to include in each batch when computing batch means and lag-1 batch autocorrelations. The batch size has the single largest impact on the time required to compute summary statistics. The computation time is decreased dramatically as the batch size increases. Consequently, users may want to increase the value of this variable for long MCMC sequences.

dev = «see below» Character string giving the name of the function that creates graphics windows on the current platform. For Unix systems this is either "motif", "openlook", or "X11". The default is "motif" for the UNIX S-PLUS, "win.graph" for Windows S-PLUS, "X11" for UNIX R, and "windows" for Windows R.

dev.list = numeric(0) Numeric vector containing the active graphics windows in use by the program. This is automatically maintained by the program, user should not directly modify this variable.

gandr.bins = 20 Number of line segments within the MCMC sequence at which to plot the Gelman and Rubin shrink factors.

gandr.win = 0.50 Proportion of iterations to include in the Brooks, Gelman, and Rubin Statistics.

geweke.bins = 10 Number of line segments within the MCMC sequence at which to plot the Geweke convergence diagnostics.

geweke.first = 0.1 Proportion of iterations to include in the first window when computing the Geweke convergence diagnostics.

geweke.last = 0.5 Proportion of iterations to include in the last window when computing the Geweke convergence diagnostics.

handw.error = 0.1 Accuracy of the posterior estimates when computing the Heidleberger and Welch convergence diagnostics.

kernel = "gaussian" Character string giving the type of window used in estimating the density functions for parameters. Other choices are "cosine", "rectangular", or "triangular".

legend = TRUE Logical value indicating that a legend be included in the plots.

path = "" Character string giving the directory path in which the raw data files are stored. The default may be used if the files are located in the current working directory. The specified path should not end with a slash(es).

plot.mfdim = c(3, 2) Numeric vector giving the maximum number of rows and columns, respectively, of plots to include in a single graphics window.

plot.new = F Logical value indicating that a new graphics window be automatically opened upon successive calls to boa.plot(). Otherwise, previous graphics windows will be closed.

plot.onelink = FALSE Logical value indicating that each plot should include only one MCMC sequence. Otherwise, all sequences are displayed on the same plot.

quantiles = c(0.025, 0.5, 0.975) Vector of probabilities at which to compute the quantiles. Values must be between 0 and 1.

randl.error = 0.005 Desired amount of error in estimating the quantile specified in the Raftery and Lewis convergence diagnostics.

randl.delta = 0.001 Delta valued used in computing the Raftery and Lewis convergence diagnostics.

randl.q = 0.025 Quantile to be estimated in computing the Raftery and Lewis convergence diagnostics.

title = TRUE Logical value indicating that a title be added to the plots.

Side Effects

When variables are set, boa.par() modifies the internal list .boa.par. If boa.par() is called with either a list as the single argument, or with one or more arguments in the <name> = <value> form, the variables specified by the names in the arguments are modified.

Author(s)

Brian J. Smith

`boa.pardesc` *Global Parameters Descriptions*

Description

Returns descriptive information on the global parameters.

Usage

```
boa.pardesc()
```

Value

A character matrix whose rows and columns ("group", "method", "desc", "par", "note") contain the global parameters and the corresponding descriptors (group = "Analysis", "Data", or "Plot"; method = subgroup classification; desc = parameter description; par = parameter name name, note = information concerning the possible values for the parameter).

Author(s)

Brian J. Smith

See Also

`boa.par`

`boa.plot` *Plot Parameters in MCMC Sequences*

Description

Automatically generates the specified plot type for all parameters in the working session list of MCMC sequences. This function takes care of all the plotting tasks - opening windows, setting the number of plots per page, and adding titles.

Usage

```
boa.plot(type, dev = boa.par("dev"), mfdim = boa.par("plot.mfdim"),
newplot = boa.par("plot.new"), onelink = boa.par("plot.onelink"),
title = boa.par("title"))
```

Arguments

`type`	The type of plots to generate. The supported types are: • "acf" = autocorrelation functions • "bandg" = Brooks and Gelman multivariate shrink factors • "density" = density functions • "gandr" = Gelman and Rubin shrink factors • "geweke" = Geweke convergence diagnostics • "history" = running means • "trace" = trace histories
`dev`	Character string giving the name of the function that creates graphics windows on the current platform. For Unix systems this is either "motif", "openlook", or "X11". For windows, this should be set to "win.graph"
`mfdim`	Numeric vector giving the maximum number of rows and columns, respectively, of plots to include in a single graphics window. If only one graphics window is opened, 'mfdim' is proportionately scaled down so as to minimize the number of empty frames within that window.
`newplot`	Logical value indicating that a new graphics window be automatically opened. Otherwise, previous graphics windows will be closed.
`onelink`	Logical value indicating that each plot should include only one MCMC sequence. Otherwise, all sequences are displayed on the same plot.
`title`	Logical value indicating that a title be added to the plot.

Value

A logical value indicating that the plots were successfully created.

Author(s)

Brian J. Smith

See Also

boa.plot.acf, boa.plot.bandg, boa.plot.density, boa.plot.gandr, boa.plot.geweke, boa.plot.history, boa.plot.trace

---

`boa.plot.acf` *Plot Autocorrelation Function*

Description

Creates a single plot of the lag autocorrelations for a specified parameter.

Usage

```
boa.plot.acf(lname, pname, annotate = boa.par("legend"))
```

Arguments

`lname`	Character string giving the name of the desired MCMC sequence in the working session list of sequences.
`pname`	Character string giving the name of the parameter to be plotted.
`annotate`	Logical value indicating that a legend be included in the plot.

Value

A logical value indicating that the plot was successfully created.

Author(s)

Brian J. Smith

See Also

boa.acf, boa.print.acf

`boa.plot.bandg` *Plot Brooks and Gelman Multivariate Shrink Factors*

Description

Plots the Brooks and Gelman multivariate shrink factors within different line segments across the MCMC sequences. This diagnostic is a multivariate extension to the Gelman and Rubin shrink factors.

Usage

```
boa.plot.bandg(bins = boa.par("gandr.bins"), win = boa.par("gandr.win"),
annotate = boa.par("legend"))
```

Arguments

`bins`	Number of line segments within the MCMC sequence at which to plot the Gelman and Rubin shrink factors. The first segment contains the first 50 iterations; the remaining iterations are partitioned into equal bins and added incrementally to construct the remaining line segments. The shrink factors are plotted against the maximum iteration number for the segment. Cubic splines are used to interpolate through the point estimates for each segment.
`win`	Proportion of iterations to include in the Brooks, Gelman, and Rubin Statistics.
`annotate`	Logical value indicating that a legend be included in the plot.

Value

A logical value indicating that the plot was successfully created.

Author(s)

Brian J. Smith

See Also

`boa.chain.gandr`, `boa.plot`, `boa.plot.gandr`, `boa.print.gandr`

boa.plot.close *Close Graphics Windows*

Description

Closes graphics windows opened with boa.plot.open().

Usage

```
boa.plot.close(which = dev.cur())
```

Arguments

which	Integer vector specifying the graphics windows to be closed.

Value

If the specified graphics windows are successfully closed, the number of the current graphics window is returned. Otherwise NULL is returned.

Author(s)

Brian J. Smith

See Also

boa.plot.open

boa.plot.density *Plot Density Functions*

Description

Estimates and displays, in a single plot, the density function(s) for the specified parameter(s).

Usage

```
boa.plot.density(lnames, pname, bandwidth = boa.par("bandwidth"),
window = boa.par("kernel"), annotate = boa.par("legend"))
```

Arguments

lnames	Character vector giving the names of the desired MCMC sequence in the working session list of sequences.
pname	Character string giving the name of the parameter to be plotted.
bandwidth	Function for computing the bandwidth used in estimating the density functions for parameters. This should take one argument which is a numeric vector of data on which density estimation is to be performed. A constant bandwidth may be specified by having this function return the desired constant.
window	Character string giving the type of window used in estimating the density functions for the parameters. Available choices are "cosine", "gaussian", "rectangular", or "triangular".
annotate	Logical value indicating that a legend be included in the plot.

Value

A logical value indicating that the plot was successfully created.

Author(s)

Brian J. Smith

See Also

boa.plot

---

boa.plot.gandr *Plot Gelman and Rubin Shrink Factors*

Description

Creates a single plot of the Gelman and Rubin shrink factors within different line segments for a parameter in an MCMC sequence.

Usage

```
boa.plot.gandr(pname, bins = boa.par("gandr.bins"), alpha = boa.par("alpha"),
win = boa.par("gandr.win"), annotate = boa.par("legend"))
```

Arguments

pname	Character string giving the name of the parameter in the working session list of MCMC sequences to be plotted.

bins	Number of line segments within the MCMC sequence at which to plot the Gelman and Rubin shrink factors. The first segment contains the first 50 iterations; the remaining iterations are partitioned into equal bins and added incrementally to construct the remaining line segments. The shrink factors are plotted against the maximum iteration number for the segment. Cubic splines are used to interpolate through the point estimates for each segment.
alpha	Quantile (1 - alpha / 2) at which to estimate the upper limit of the shrink factor.
win	Proportion of iterations to include in the Brooks, Gelman, and Rubin Statistics.
annotate	Logical value indicating that a legend be included in the plot.

Value

A logical value indicating that the plot was successfully created.

Author(s)

Brian J. Smith

See Also

boa.chain.gandr, boa.plot, boa.plot.bandg, boa.print.gandr

---

boa.plot.geweke *Plot Geweke Convergence Diagnostics*

Description

Creates a single plot of the Geweke convergence diagnostics within different line segments for a parameter in an MCMC sequence.

Usage

```
boa.plot.geweke(lname, pname, bins = boa.par("geweke.bins"), p.first = boa.par("gew
```

Arguments

lname	Character string giving the name of the desired MCMC sequence in the working session list of sequences.
pname	Character string giving the name of the parameter to be plotted.
bins	Number of line segments within the MCMC sequence at which to plot the Geweke convergence diagnostics. The ith line segment contains the last ((bins - i + 1) / bins)*100 bins. This may lead to segments in which there are too few iterations to compute the convergence diagnostics. Such segments, if they exist, are automatically omitted from the calculations.
p.first	Proportion of iterations to include in the first window.
p.last	Proportion of iterations to include in the last window.
alpha	Alpha level for the rejection region lines drawn on the graph.
annotate	Logical value indicating that a legend be included in the plot.

Value

A logical value indicating that the plot was successfully created.

Author(s)

Brian J. Smith, Nicky Best, Kate Cowles

See Also

boa.geweke, boa.plot, boa.print.geweke

---

boa.plot.history *Plot Parameter Estimation Histories*

Description

Computes and displays, in a single plot, the running mean(s) for the specified parameter(s).

Usage

```
boa.plot.history(lnames, pname, annotate = boa.par("legend"))
```

Arguments

lnames	Character vector giving the name of the desired MCMC sequence in the working session list of sequences.
pname	Character vector giving the names of the parameters to be plotted.
annotate	Logical value indicating that a legend be included in the plot.

Value

A logical value indicating that the plot was successfully created.

Author(s)

Brian J. Smith

See Also

boa.plot

boa.plot.open *Open Graphics Windows*

Description

Opens a graphics window.

Usage

```
boa.plot.open()
```

Value

The number of the current graphics window is returned.

Author(s)

Brian J. Smith

See Also

boa.plot.close

boa.plot.par *Set Plotting Parameters*

Description

Set Plotting Parameters

Usage

```
boa.plot.par(mfdim = c(1, 1), title = TRUE)
```

Arguments

mfdim	Numeric vector with two elements giving the number of rows and column, respectively, of plots to display on the page. Plots will be drawn row-by-row.
title	Logical value indicating that a title be added to the plot.

Author(s)

Brian J. Smith

boa.plot.title *Add Plot Title*

Description

Adds a title to the graphics window. This function should be called after all plots have been added to the target graphics window.

Usage

```
boa.plot.title(text)
```

Arguments

text	Character string giving the title to be centered and displayed across the top of the graphics window.

Author(s)

Brian J. Smith

boa.plot.trace *Plot Trace Histories*

Description

Displays, in a single plot, the trace histories for the specified parameter(s).

Usage

```
boa.plot.trace(lnames, pname, annotate = boa.par("legend"))
```

Arguments

lnames	Character vector giving the name of the desired MCMC sequence in the working session list of sequences.
pname	Character string giving the name of the parameters to be plotted.
annotate	Logical value indicating that a legend be included in the plot.

Value

A logical value indicating that the plot was successfully created.

Author(s)

Brian J. Smith

See Also

boa.plot

---

boa.pnames *Extract Parameter Names*

Description

Extracts the parameter names in an MCMC sequence.

Usage

```
boa.pnames(link)
```

Arguments

link	Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to dimnames(link).

Value

A character vector containing the parameter names in 'link'.

Author(s)

Brian J. Smith

See Also

boa.getparms, boa.iter

---

boa.print.acf *Print Autocorrelation Functions*

Description

Iteratively calls boa.acf() to display the lag autocorrelations for the parameters in each of the MCMC sequences in the working session list of sequences.

Usage

```
boa.print.acf(lags = boa.par("acf.lags"))
```

Arguments

lags	Numeric vector of lags at which to estimate the autocorrelation functions.

Author(s)

Brian J. Smith

See Also

boa.acf, boa.plot, boa.plot.acf

---

boa.print.cor *Print Correlation Matrices*

Description

Iteratively computes and displays the correlation matrices for the parameters in each of the MCMC sequences in the working session list of sequences.

Usage

```
boa.print.cor()
```

Author(s)

Brian J. Smith

---

boa.print.gandr *Print Gelman and Rubin Convergence Diagnostics*

Description

Calls boa.chain.gandr() and displays the Gelman and Rubin convergence diagnostics for the MCMC sequences in the working session list of sequences.

Usage

```
boa.print.gandr(alpha = boa.par("alpha"), win = boa.par("gandr.win"))
```

Arguments

alpha	Quantile (1 - alpha / 2) at which to estimate the upper limit of the corrected shrink factor.
win	Proportion of iterations to include in the Brooks, Gelman, and Rubin Statistics.

Author(s)

Brian J. Smith

See Also

boa.chain.gandr, boa.plot, boa.plot.bandg, boa.plot.gandr

---

boa.print.geweke *Print Geweke Convergence Diagnostics*

Description

Iteratively calls boa.geweke() to display the Geweke convergence diagnostics for the parameters in each of the MCMC sequences in the working session list of sequences.

Usage

```
boa.print.geweke(p.first = boa.par("geweke.first"), p.last = boa.par("geweke.last")
```

Arguments

p.first	Proportion of iterations to include in the first window.
p.last	Proportion of iterations to include in the last window.

Author(s)

Brian J. Smiht

See Also

boa.geweke, boa.plot, boa.plot.geweke

---

boa.print.handw *Print Heidelberger and Welch Convergence Diagnostics*

Description

Iteratively calls boa.handw() to display the Heidelberger and Welch convergence diagnostics for the parameters in each of the MCMC sequences in the working session list of sequences.

Usage

```
boa.print.handw(error = boa.par("handw.error"), alpha = boa.par("alpha"))
```

Arguments

error	Accuracy of the posterior estimates for the monitored parameters.
alpha	Alpha level for the confidence in the sample mean of the retained iterations.

Author(s)

Brian J. Smith

See Also

boa.handw

---

boa.print.hpd *Print Highest Probability Density Intervals*

Description

Iteratively calls boa.hpd() to display the highest probability density (HPD) intervals for the parameters in each of the MCMC sequences in the working session list of sequences.

Usage

```
boa.print.hpd(alpha = boa.par("alpha"))
```

Arguments

alpha	Specifies the 100*(1-alpha)% HPD intervals to be computed.

Author(s)

Brian J. Smith

See Also

boa.hpd

boa.print.info *Print Information about MCMC Sequences*

Description

Iteratively calls boa.chain.info() to display information from the MCMC sequences in the specified session list of sequences.

Usage

```
boa.print.info(which = "work")
```

Arguments

which	Character string specifying which session list of MCMC sequences for which to print information. The two choices are "work" (used in all analyses) or "data" (the template for "work").

Value

The full list obtained from boa.chain.info() is returned, but not printed.

Author(s)

Brian J. Smith

See Also

boa.chain.info

boa.print.par *Print Global Parameters' Descriptions and Current Values*

Description

Displays a description of and the current value for all global parameters.

Usage

```
boa.print.par(group)
```

Arguments

group	Character string specifying which parameter group to display. The groups are "Analysis", "Data", and "Plot". If omitted, information on all global parameters is displayed.

Value

The character matrix obtained from boa.pardesc() is returned, but not printed.

Author(s)

Brian J. Smith

See Also

boa.pardesc

---

boa.print.randl *Print Raftery and Lewis Convergence Diagnostics*

Description

Iteratively calls boa.randl() to display the Raftery and Lewis convergence diagnostics for the parameters in each of the MCMC sequences in the working session list of sequences.

Usage

```
boa.print.randl(q = boa.par("randl.q"), error = boa.par("randl.error"), prob = 1 -
```

Arguments

q	Quantile to be estimated.
error	Desired amount of error in estimating the specified quantile 'q'.
prob	Probability of attaining the desired degree of 'error'.
delta	Delta value used in computing the convergence diagnostic.

Author(s)

Brian J. Smith

See Also

boa.randl

boa.print.stats *Print Summary Statistics*

Description

Iteratively calls boa.stats() to display summary statistics for the parameters in each of the MCMC sequences in the working session list of sequences.

Usage

```
boa.print.stats(probs = boa.par("quantiles"), batch.size = boa.par("batch.size"))
```

Arguments

probs	Vector of probabilities at which to compute the quantiles. Values must be between 0 and 1.
batch.size	Number of iterations to include in each batch when computing batch means and lag-1 batch autocorrelation function.

Author(s)

Brian J. Smith

See Also

boa.stats

boa.quit *Quit Session*

Description

boa.quit() is called to end a BOA session. It removes the objects used to store the session lists of MCMC sequences and the global parameters. Failure to do this may result in lost or lingering data.

Usage

```
boa.quit()
```

Author(s)

Brian J. Smith

See Also

boa.save, boa.init

`boa.randl` *Raftery and Lewis Convergence Diagnostics*

Description

Computes the Raftery and Lewis convergence diagnostics for the parameters in an MCMC sequence.

Usage

```
boa.randl(link, q, error, prob, delta)
```

Arguments

`link`	Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to `dimnames(link)`.
`q`	Quantile to be estimated.
`error`	Desired amount of error in estimating the specified quantile 'q'.
`prob`	Probability of attaining the desired degree of error - 'error'.
`delta`	Delta value used in computing the convergence diagnostics.

Value

A matrix whose columns and rows are the Raftery and Lewis convergence diagnostics (i.e. thin, burn-in, total, lower bound, and dependence factor) and the monitored parameters, respectively.

Author(s)

Brian J. Smith, Nicky Best, Kate Cowles

References

1. Raftery, A. L. and Lewis, S. (1992a). Comment: One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo. Statistical Science, 7, 493-7.
2. Raftery, A. L. and Lewis, S. (1992b). How many iterations in the Gibbs sampler? In Bayesian Statistics 4, (ed. J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith), pp. 763-74. Oxford University Press.

See Also

`boa.print.randl`

boa.save *Save Session*

Description

Save the current state of the session lists of MCMC sequences and the global parameters to a database object.

Usage

```
boa.save(name, envir = globalenv())
```

Arguments

name	Character string giving the name of the object to which the current session should be saved.
envir	The 'environment' to which the object should be saved. For more information, consult the help documentation in R on the assign() function.

Value

A logical value indicating that the session was successfully saved to the specified object.

Author(s)

Brian J. Smith

boa.sortiter *Sort by Iteration Numbers*

Description

Sorts an MCMC sequence by increasing iteration numbers.

Usage

```
boa.sortiter(link)
```

Arguments

link	Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to `dimnames(link)`.

Value

A copy of 'link' with the rows sorted by increasing iteration numbers.

Author(s)

Brian J. Smith

See Also

boa.sortparms

---

boa.sortparms *Sort by Parameter Names*

Description

Sorts the parameters of an MCMC sequence alphabetically by parameter names.

Usage

```
boa.sortparms(link)
```

Arguments

link	Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to dimnames(link).

Value

A copy of 'link' with the columns sorted alphabetically by parameter names.

Author(s)

Brian J. Smith

See Also

boa.sortiter

boa.stats *Summary Statistics*

Description

Computes summary statistics for the parameters in an MCMC sequence.

Usage

```
boa.stats(link, probs, batch.size)
```

Arguments

link	Matrix whose columns and rows contain the monitored parameters and the MCMC iterations, respectively. The iteration numbers and parameter names must be assigned to `dimnames(link)`.
probs	Vector of probabilities at which to compute the quantiles. Values must be between 0 and 1.
batch.size	Number of iterations to include in each batch when computing batch means and lag-1 batch autocorrelations.

Value

A matrix whose columns and rows contain the summary statistics (i.e. sample mean, standard deviation, naive standard error, MC error, batch standard error, lag-1 batch autocorrelation, specified quantiles, minimum and maximum iteration numbers, and total iterations in the sample) and the monitored parameters, respectively.

Author(s)

Brian J. Smith

See Also

boa.print.stats

boa.transform *Transform Variables*

Description

Applies a log or logit transformation to map bounded variables to the entire real line.

Usage

```
boa.transform(x, support, inv = FALSE)
```

Arguments

x	Numeric object to be transformed.
support	Vector giving the lower and upper limits of the support for the untransformed data. Unbounded lower or upper limits should be specified as -Inf or Inf, respectively.
inv	Logical value indicating that the inverse transformation should be applied.

Value

Transformed or inverse-transformed analogue of 'x'.

Author(s)

Brian J. Smith

---

boa.version *Version Information*

Description

Information about this release of BOA.

Usage

```
boa.version()
```

Value

name	Character string giving the program name.
major	Major number of this release.
minor	Minor number of this release.
revision	Revision number for this version.
system	Character string giving the platform for which this version of BOA was designed.

Author(s)

Brian J. Smith

See Also

boa.license

colVars *Column Variances*

Description

Computes variances by column.

Usage

```
colVars(x, na.rm = FALSE, unbiased = TRUE, SumSquares = FALSE)
```

Arguments

x	A matrix, array, or data frame
na.rm	Logical value indicating that missing values be omitted from the calculations.
unbiased	Not implemented
SumSquares	Not implemented

line *BUGS Line Example*

Description

The Line example involves a linear regression analysis of the data points (1,1), (2,3), (3,3), (4,3), and (5,5). The proposed Bayesian model is

$$y[i]\ N(mu[i], tau)$$

$$mu[i] = alpha + beta * (x[i] - mean(x[]))$$

with the following priors:

$$alpha\ N(0, 0.0001)$$

$$beta\ N(0, 0.0001)$$

$$tau\ Gamma(0.001, 0.001)$$

Two parallel chains from the MCMC sampler were generated and the output saved in the S data frames line1 and line2.

Usage

```
data(line)
```

Format

The data frames line1 and line2 each containing 200 observations.

References

Spiegelhalter, D., Thomas, A. and Best, N., (2000). WinBugs Version 1.4 User Manual.

Index

∗Topic **datasets**
line, 45
∗Topic **hplot**
boa.plot, 24
boa.plot.acf, 25
boa.plot.bandg, 26
boa.plot.density, 27
boa.plot.gandr, 28
boa.plot.geweke, 29
boa.plot.history, 30
boa.plot.trace, 32
∗Topic **interface**
boa.menu, 20
∗Topic **internal**
boa.batchMeans, 2
boa.getiter, 12
boa.getparms, 12
boa.gewekePwr, 14
boa.iter, 19
boa.license, 19
boa.pardesc, 23
boa.plot.close, 27
boa.plot.open, 31
boa.pnames, 33
boa.sortiter, 41
boa.sortparms, 42
boa.transform, 43
boa.version, 44
colVars, 45
∗Topic **print**
boa.print.acf, 33
boa.print.cor, 34
boa.print.gandr, 34
boa.print.geweke, 35
boa.print.handw, 35
boa.print.hpd, 36
boa.print.info, 37
boa.print.par, 37
boa.print.randl, 38
boa.print.stats, 39
∗Topic **univar**
boa.acf, 1
boa.chain.gandr, 6
boa.geweke, 13
boa.handw, 14
boa.hpd, 15
boa.randl, 40
boa.stats, 43
∗Topic **utilities**
boa.chain, 3
boa.chain.add, 4
boa.chain.collapse, 5
boa.chain.del, 5
boa.chain.eval, 6
boa.chain.import, 8
boa.chain.info, 9
boa.chain.reset, 10
boa.chain.subset, 10
boa.chain.support, 11
boa.importASCII, 16
boa.importBUGS, 17
boa.importMatrix, 17
boa.init, 18
boa.load, 20
boa.par, 21
boa.plot.par, 31
boa.plot.title, 32
boa.quit, 39
boa.save, 41

boa.acf, 1, *25*, *34*
boa.batchMeans, 2
boa.chain, 3
boa.chain.add, 4, *8*
boa.chain.collapse, *3*, 5, *10, 11*
boa.chain.del, 5
boa.chain.eval, 6
boa.chain.gandr, 6, *26*, *29*, *35*
boa.chain.import, *4*, 8, *16–18*

boa.chain.info, 9, *37*
boa.chain.reset, *3*, *5*, 10, *11*
boa.chain.subset, *3*, *5*, 10, *10*
boa.chain.support, 11
boa.getiter, 12, *13*, *19*
boa.getparms, 12, *12*, *33*
boa.geweke, 13, *30*, *35*
boa.gewekePwr, 14
boa.handw, 14, *36*
boa.hpd, 15, *36*
boa.importASCII, *8*, 16, *17, 18*
boa.importBUGS, *8*, *16*, 17, *18*
boa.importMatrix, *16*, 17, *17*
boa.init, 18, *39*
boa.iter, *12*, 19, *33*
boa.license, 19, *44*
boa.load, *18*, 20
boa.menu, 20
boa.par, 21, *23*
boa.pardesc, 23, *38*
boa.plot, *2*, *7*, *14*, 24, *26*, *28–30*, *33–35*
boa.plot.acf, *2*, 25, *25*, *34*
boa.plot.bandg, *7*, *25*, 26, *29*, *35*
boa.plot.close, 27, *31*
boa.plot.density, *25*, 27
boa.plot.gandr, *7*, *25, 26*, 28, *35*
boa.plot.geweke, *14*, *25*, 29, *35*
boa.plot.history, *25*, 30
boa.plot.open, *27*, 31
boa.plot.par, 31
boa.plot.title, 32
boa.plot.trace, *25*, 32
boa.pnames, *13*, *19*, 33
boa.print.acf, *2*, *25*, 33
boa.print.cor, 34
boa.print.gandr, *7*, *26*, *29*, 34
boa.print.geweke, *14*, *30*, 35
boa.print.handw, *15*, 35
boa.print.hpd, 36
boa.print.info, *9*, 37
boa.print.par, 37
boa.print.randl, 38, *40*
boa.print.stats, 39, *43*
boa.quit, *18*, 39
boa.randl, *38*, 40
boa.save, *39*, 41
boa.sortiter, 41, *42*
boa.sortparms, 42, *42*
boa.stats, *39*, 43
boa.transform, 43
boa.version, *19*, 44

colVars, 45

line, 45
line1 *(line)*, 45
line2 *(line)*, 45